(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2001 (18.01.2001)

PCT

(10) International Publication Number
WO 01/05089 A2

(51) International Patent Classification[7]: H04L 9/00

(21) International Application Number: PCT/US00/18545

(22) International Filing Date: 7 July 2000 (07.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/142,870	8 July 1999 (08.07.1999)	US
60/159,012	12 October 1999 (12.10.1999)	US
09/510,486	23 February 2000 (23.02.2000)	US

(71) Applicant *(for all designated States except US)*: **BROADCOM CORPORATION** [US/US]; 16215 Alton Parkway, Irvine, CA 92618 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KRISHNA, Suresh** [US/US]; 695 S. Knickerbocker Drive #6, Sunnyvale, CA 94087 (US). **OWEN, Christopher** [US/US]; 708 Blossom Hill Road #198, Los Gatos, CA 95032 (US).

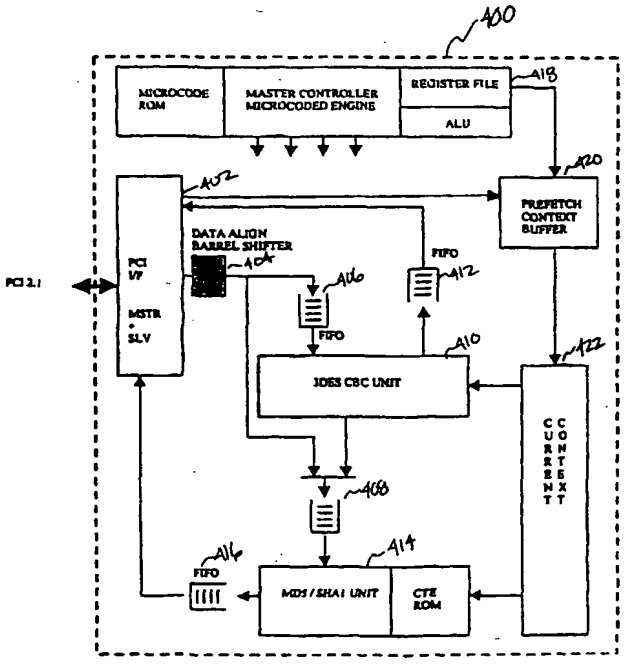
(74) Agent: **AUSTIN, James, E.**; Beyer Weaver & Thomas, LLP, P.O. Box 130, Mountain View, CA 94042-0130 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: SECURITY CHIP ARCHITECTURE AND IMPLEMENTATIONS FOR CRYPTOGRAPHY ACCELERATION



(57) Abstract: An architecture and a method for a cryptography acceleration is disclosed that allows significant performance improvements without the use of external memory. Specifically, the chip architecture enables "cell-based" processing of random-length IP packets. The IP packets, which may be of variable and unknown size, are split into fixed-size "cells". The fixed-sized cells are then processed and reassembled into packets. The cell-based packet processing architecture of the present invention allows the implementation of a processing pipeline that has known processing

[Continued on next page]

WO 01/05089 A2

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *Without international search report and to be republished upon receipt of that report.*

throughput and timing characteristics, thus making it possible to fetch and process the cells in a predictable time frame. The architecture is scalable and is also independent of the type of cryptography performed. The cells may be fetched ahead of time (pre-fetched) and the pipeline may be staged in such a manner that attached (local) memory is not required to store packet data or control parameters.

SECURITY CHIP ARCHITECTURE AND IMPLEMENTATIONS FOR CRYPTOGRAPHY ACCELERATION

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to the field of cryptography, and more particularly to an architecture and method for cryptography acceleration.

2. Description of the Related Art

Many methods to perform cryptography are well known in the art and are discussed, for example, in Applied Cryptography, Bruce Schneier, John Wiley & Sons, Inc. (1996, 2nd Edition), herein incorporated by reference. In order to improve the speed of cryptography processing, specialized cryptography accelerator chips have been developed. For example, the Hi/fn ™ 7751 and the VLSI ™ VMS115 chips provide hardware cryptography acceleration that out-performs similar software implementations. Cryptography accelerator chips may be included in routers or gateways, for example, in order to provide automatic IP packet encryption/decryption. By embedding cryptography functionality in network hardware, both system performance and data security are enhanced.

However, these chips require sizeable external attached memory in order to operate. The VLSI VMS118 chip, in fact, requires attached synchronous SRAM, which is the most expensive type of memory. The additional memory

requirements make these solutions unacceptable in terms of cost versus performance for many applications.

Also, the actual sustained performance of these chips is much less than peak throughput that the internal cryptography engines (or "crypto engines") can sustain. One reason for this is that the chips have a long "context" change time. In other words, if the cryptography keys and associated data need to be changed on a packet-by-packet basis, the prior art chips must swap out the current context and load a new context, which reduces the throughput. The new context must generally be externally loaded from software, and for many applications, such as routers and gateways that aggregate bandwidth from multiple connections, changing contexts is a very frequent task.

Recently, an industry security standard has been proposed that combines both "DES/3DES" encryption with "MD5/SHA1" authentication, and is known as "IPSec." By incorporating both encryption and authentication functionality in a single accelerator chip, over-all system performance can be enhanced. But due to the limitations noted above, the prior art solutions do not provide adequate performance at a reasonable cost.

Thus it would be desirable to have a cryptography accelerator chip architecture that is capable of implementing the IPSec specification (or any other cryptography standard), that does not require external memory, and that can change context information quickly.

SUMMARY OF THE INVENTION

In general, the present invention provides an architecture for a cryptography accelerator chip that allows significant performance improvements

over previous prior art designs. Specifically, the chip architecture enables "cell-based" processing of random-length IP packets. The IP packets, which may be of variable and unknown size, are split into smaller fixed-size "cells." The fixed-sized cells are then processed and reassembled into packets. For example, the incoming IP packets may be split into 64-byte cells for processing.

The cell-based packet processing architecture of the present invention allows the implementation of a processing pipeline that has known processing throughput and timing characteristics, thus making it possible to fetch and process the cells in a predictable time frame. The present architecture is scalable and is also independent of the type of cryptography performed. In preferred embodiments, the cells may be fetched ahead of time (pre-fetched) and the pipeline may be staged in such a manner that attached (local) memory is not required to store packet data or control parameters.

In a first embodiment, an IPSec processing chip may be implemented by having 3DES-CBC and MD5/SHA1 processing blocks. The processing of the cells is pipelined and the sequencing is controlled by a programmable microcontroller. In a second embodiment, Diffie-Hellman or RSA and DSA public key processing may be added as well. Additional processing blocks may be implemented as well. The present invention provides a performance improvement over the prior art designs, without requiring any additional external memory.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

Figure 1 is a high-level block diagram of a system implementing a cryptography accelerator chip according to the present invention;

Figure 2 is a high-level block diagram of a cryptography accelerator chip of the present invention;

Figure 3 is a diagram illustrating the conversion of a random-length packet to a fixed-size cell, as disclosed by the present invention;

Figure 4 is a block diagram of a cryptography accelerator chip configured according to a first embodiment of the present invention;

Figure 5 is a block diagram of a cryptography accelerator chip configured according to a second embodiment of the present invention;

Figure 6 is a block diagram illustrating the packet processing of the present invention;

Figures 7(A) - 7(D) are flowcharts illustrating one embodiment of the packet sequencing procedure of the present invention; and

Figure 8 is a graph comparing the performance of a cryptography accelerator chip configured according to the present invention with a prior-art cryptography accelerator chip.

DETAILED DESCRIPTION OF THE INVENTION

The following description is provided to enable any person skilled in the art to make and use the invention and sets forth the best modes contemplated by the inventors for carrying out the invention. Various modifications, however, will remain readily apparent to those skilled in the art, since the basic principles of the present invention have been defined herein specifically to provide an architecture and method for cryptography acceleration.

In general, the present invention provides an architecture for a cryptography accelerator chip that allows significant performance improvements over previous prior art designs. Specifically, the chip architecture enables "cell-based" processing of random-length IP packets. Cell-based packet processing involves the splitting of IP packets, which may be of variable and unknown size, into smaller fixed-size "cells." The fixed-sized cells are then processed and reassembled (recombined) into packets. For example, the incoming IP packets may be split into 64-byte cells for processing. The cell-based packet processing architecture of the present invention allows the implementation of a processing pipeline that has known processing throughput and timing characteristics, thus making it possible to fetch and process the cells in a predictable time frame. In preferred embodiments, the cells may be fetched ahead of time (pre-fetched) and the pipeline may be staged in such a manner that attached (local) memory is not required to store packet data or control parameters.

At present, the other known solutions in the industry use a completely different architecture that relies on having local, attached memory to hold context information, packet data or both, depending on the architecture. Thus, the prior-art designs require an external block of memory very close to the chip. The present invention does not require external memory due to the highly pipelined design that breaks up packets into fixed-sized cells. Because the cells are of a fixed size, the present invention can pre-fetch the fixed-sized cells ahead of time before processing.

The size of memory required on an accelerator chip that is constructed according to the present invention is substantially less than the memory required for other solutions. The present invention only needs enough memory on chip to

store a few 64-byte cells, context information, keys, etc for two or more packets, and is able to sustain full performance for any packet size and any number of contexts. The prior art solutions require sufficient memory to hold the context for several hundred to several thousand different packets, plus enough memory to hold the packet data as well.

The cell based pipeline mechanism of the present invention is driven by a programmable control processor, which is responsible for sequencing the flow of packets as well as control information through the set of processing units. The control processor determines at any given time how packets are split up into fixed sized cells, and when the cells should be read from the bus into the on-chip cell memory. The processor also pre-fetches context information that describes what processing needs to be applied to the packets. The control processor then sequences the computational units (crypto, authentication, compression, etc.) to apply the specific algorithms specified in the context information to the cells that have been previously fetched. The control processor also writes out the processed result in cell size "chunks." Furthermore, the present invention supports a mode in which consecutive back-to-back packets can use different keys and different cryptographic formats without any loss of performance because the context and the packet data are pre-fetched.

In standard networks, IP packets can be of variable size, anywhere from 1 to 2^{16} bytes, although in practice most packets are between 64 and 8k bytes. According to an embodiment of the present invention, the variable-length packets are split into standard 64-byte cells, although other implementations may use a different fixed-size for each cell. The present invention relies on the control processor to efficiently sequence the cell processing. For example, if there are

many large packets back-to-back, the control processor focuses on processing the current packet as quickly as possible, but if there are some small packets queued up, it will emphasize the pre-fetch of the control information, since this will be the likely processing bottleneck.

Under certain circumstances, the processor may decide not to pre-fetch the next set of context and key information, and in certain cases it will pre-fetch this information. For example, the processor may decide not to pre-fetch under the following two scenarios:

1) if the system bus is currently heavily loaded with processing data from the current packet (such as writing back processed cells), the control processor would delay any further pre-fetch requests to avoid overloading the system bus, until the system bus is available;

2) if the control processor itself is busy processing control information for the current packet, such as fetching new cells for the current packet, then the control processor will delay the pre-fetch of the next set of packets and associated control information.

Since the control processor can be programmed via microcode instructions, the architecture can be implemented in a relatively small die size relative to the performance levels that can be achieved, which is a big advantage over competing solutions. The architecture of the present invention is also independent of the type of crypto engines used, and therefore new algorithms can be supported simply be adding additional crypto blocks.

As shown in Figure 1, the present invention may be implemented as a stand-alone cryptography accelerator chip 102 and incorporated into a standard processing system 100. The cryptography accelerator chip 102 may be connected

to a standard PCI bus 104 via a standard on-chip PCI interface. The processing system 100 includes a processing unit 106 and a system memory unit 108. The processing unit 106 and the system memory unit 108 may be attached to the system bus 104 via a bridge and memory controller 110. A LAN interface 114 attaches the processing system 100 to a local area network and receives packets for processing and writes out processed packets to the network. Likewise, a WAN interface 112 connects the processing system to a WAN, such as the Internet, and manages in-bound and out-bound packets, providing automatic security processing for IP packets.

Figure 2 is a high-level block diagram of the cryptography chip architecture of the present invention. A standard PCI interface 202 provides a standard interface for connecting the chip 200 to external systems. According to this embodiment, the PCI bus is a 32-bit bus operating at up to 33 MHz. Of course, other interfaces and configurations may be used, as is well known in the art, without departing from the scope of the present invention. The IP packets are read into a FIFO (First In First Out buffer) 204, where the random-length packets are split into fixed-sized cells. The fixed-sized cells are then stored in payload cell buffers 210 via the internal bus 224. Context buffers 208 store "context" information for the associated fixed-sized cells, such as encryption keys, data, etc. A programmable processor 212 controls the sequencing and processing of the fixed-sized cells, and optimizes the pipelined processing. The processor 212 is programmed via on-chip microcode stored in a microcode storage unit 214.

The fixed-sized cells are then processed in a pipelined fashion by one of the "crypto" engines. For example, the crypto engines may include "3DES-CBC/DES X" encryption/decryption 216, "MD5/SHA1" authentication/digital

signature processing 218, and compression/decompression processing 220. Note that the present architecture is independent of the types of cryptography processing performed, and additional crypto engines may be incorporated to support other current or future cryptography algorithms 222. The output cells are then stored in an output FIFO 206, in order to write the packets back out to the system via the PCI bus.

As previously discussed, the present architecture converts random-length packets into fixed-sized cells, in order to facilitate pipelined processing. This conversion is illustrated in Figure 3. Once a random-length IP packet is obtained from the system, the packet is converted into a plurality of fixed-size cells (or one cell if the packet is smaller than the standard fixed-sized cell). Since the cells have a uniform size, the processing flow can be designed to maximize the throughput of the system by incorporating pipelining design techniques, such as pre-fetching. If an IP packet is less than the standard fixed-sized cell, the packet is converted into a single fixed-sized cell and processed. The step of "recombining" in this case simply comprises converting the single cell back to an IP packet.

A first embodiment of the present invention is shown in more detail in Figure 4. An IPSec cryptography accelerator chip 400 constructed according to the present invention reads and writes data to the system via a PCI interface 402. Each incoming packet is sub-divided into fixed-size cells by a data align barrel shifter 404, wherein each cell in this implementation is 64 bytes. The data align barrel shifter 404 serves as a packet splitting unit to divide the incoming packets into fixed-sized cells. The input packets may also be scattered all over memory (i.e. fragmentation), and the data align barrel shifter unit 404 reassembles those pieces and produces as output fixed size 64 byte cells.

The size of each cell may be larger or smaller, depending on the cost and performance requirements for a given implementation. Also, other techniques may be used to sub-divide the incoming packets, as are well known in the art, without departing from the scope of the present invention. The choice of 64-byte fixed-sized cells is a design trade-off between the amount of memory needed on chip and the higher the performance that can be achieved with larger sized cells. For current cost versus performance, a 64-byte cell size is a good trade-off. Also, a 64-byte cell size is a good match for the size requirements for some of the crypto algorithms, particularly MD5/SHA1, which prefers to see 64-byte "chunks" of data.

As an incoming packet is sub-divided, the fixed-sized cells are stored in FIFO buffers 406, 408 waiting for processing by the crypto engines 410, 414. Context information needed to process the current packet is also read in and stored in the pre-fetch context buffer 420. This implementation is designed to provide industry-standard IETF IPSec encryption and authentication acceleration and therefore only includes two crypto engines. A "3DES-CBC" unit 410 is included for providing encryption and decryption of incoming packets and a "MD5/SHA1" unit 414 provides authentication and digital signature processing. For in-bound packets, the cells are first authenticated and then decrypted in parallel fashion. For out-bound packets, the cells are first encrypted then authenticated, again in pipelined fashion. The processing units 410, 414 processes the cells in the FIFOs 406, 408 using the current packet context information stored in the current context buffer 422.

The outputs of the processing units 410, 414 are stored in output FIFOs 412, 416 until the data can be written back out to system memory via the PCI

interface 402. The sequencing of the data processing and pre-fetching is controlled by the microcontroller 418, and the program code (described below) ensures that the crypto engines are continually provided with cells and context information. Since the crypto units do not have to wait while entire packets of varying sizes are read in from system memory, this procedure increases the throughput of the chip, as compared to the prior art designs. For this basic design with an internal clock speed of 60 MHz, the engine throughput is about 195 Mb/s with 3DES encryption and MD5/SHA1 authentication enabled.

This implementation is suitable for a variety of cost-sensitive applications, such as cable modems, xDSL devices, security gateways, and PC-based security accelerators. Since the present invention does not require any external memory, the cost is much lower than competing designs that require external memory. Also, testing has shown that full performance can be maintained independent of any reasonable PCI bus latency or clock frequency, since the data is pre-fetched well before it is needed for internal processing.

The interface between the cryptography accelerator chip and the host CPU software provides autonomous chip operation via an intelligent, descriptor-based DMA interface that minimizes the software-processing load. Specifically, packet data copying is avoided under all conditions. Input packet fragmentation is supported (at an IP level as well as in terms of memory allocation for the packet data) and the input fragments can be of any size (down to one byte), and can be aligned on any byte boundary. Output packet fragmentation (at an IP level as well as in terms of memory allocation for packet data) is also supported. The output fragment size can be controlled in one of two configurable ways: through a length field with each output data descriptor, or through a global output data buffer

length field. This provides the flexibility of using a fixed output fragment size, or of setting fragment size on a per-packet basis. In the present embodiment, output fragments must be aligned on 32-bit word boundaries, and must be multiples of a 32-bit word in size.

The host CPU queues up any number of packets in system memory, and passes a pointer to a master command structure that identifies these packets to the accelerator chip. The master command record is used to hand off a number of packets to the chip for processing. The structure is variable-length, and contains up to 2^{16} - 1 sets of fields, wherein each field describes one packet. This degree of flexibility allows the host CPU to queue up any number of packets, and to initiate hardware processing of all the queued packets via a single PCI write. The accelerator chip then processes all the packets as specified, returns status information to the CPU via a "done" flag, and if enabled, via an interrupt per packet, or upon global completion of all packets within a master command structure.

A unique processing context structure is associated with each packet in the master command record, which allows various packets to be processed differently even though they are all part of a common master command structure. In addition, data from each packet can be fragmented on input ("gather" function support) and on output ("scatter" function support).

A second embodiment of the present invention is illustrated in Figure 5. This embodiment is similar to the embodiment of Figure 4, except that it also includes a DH(Diffie-Hellman)/RSA/DSA unit 506, and a random number generator unit 508 to facilitate the public key processing. With an internal clock of 75 MHz, the engine throughput in this embodiment is over 400 Mb/s, with

3DES encryption and MD5/SHA1 authentication enabled. In this embodiment the PCI bus is a 64-bit bus operating at up to 66 MHz. Note that the speed of the PCI bus clock (33 MHz vs. 66 MHz) and the bus latency have very little effect on the performance of the present invention, since the accelerator chips aggressively pre-fetch and write back descriptors, command buffers, context parameters and packet data. This enables the accelerator chips to run the crypto and authentication engines at full potential despite other system latencies.

The key setup execution unit 506 accelerates the public key operations and the random number generator unit 508 generates secure private keys. Additionally, a register block 504 has 1024-bit register files to hold the large public key data used in public key processing. Although not shown in Figure 5, this embodiment includes the FIFOs and the data align barrel shifter described with reference to Figure 4. In addition to the crypto units shown, any other current or future algorithms may be supported using similar techniques.

The embodiment of Figure 5 generates SSL session keys using RSA in the following stages:

1. fetch the command context including keys and message through DMA

2. if the required operation is private key encryption, use the private key RSA algorithm with pre-computed components generated using the Chinese Remainder Theorem

3. if the required operation is public key encryption, use the public RSA algorithm

4. write the decrypted/encrypted message to the output buffer.

Alternatively, the second embodiment generates keys using the Diffie-Hellman algorithm for an IPSec session during IKE handshake according to the following stages:

1. fetch the command context and message through DMA

2. if the required operation is to generate a message to another party (g^x mod n), generate a 180-bit random number from the random number generator unit 508 and then perform the modular exponentiation with the generated random number as the exponent

3. if the required operation is to generate the shared key from the message received (Y^x mod n), perform the modular exponentiation with a previously generated random number (the random number will be a part of the command context through the DMA)

4. write the output including the random number, if applicable, to the output buffer.

Authentication using DSA algorithm for an IPSec session during IKE handshake is preformed in the following stages:

1. fetch the command context and message through DMA

2. if the required operation is to sign a message, generate a random number and compute "r" and "s" values using the SHA1 512 and key setup 506 execution units

3. if the required operation is to verify a signature, compute "v" value using SHA1 512 and key setup 506 execution units

4. write the output to the output buffer.

Figure 6 illustrates a high-level view of packet processing according to the present invention. Note that multiple sets of input packets can be specified via a

single command descriptor (i.e. a single PCI write). IPSec packets are processed in the following stages:

1. fetch the command context and data via descriptors

2. if a packet is inbound, authenticate then decrypt the cells in parallel fashion

3. if a packet is outbound, encrypt then authenticate the cells in pipelined fashion

4. write (via descriptors) the output data and authentication codes, if applicable

The command, data descriptor, packet data and context data fetch phases are completely overlapped with the engine processing. Output packet data write-back is completely overlapped as well.

The processing sequence control for the first embodiment of the present invention will now be described in further detail with reference to Figures 7(A) - 7(D). The processing has been designed to maximize the over-all chip throughput by pipelining the various functions.

The procedure disclosed in Figures 7(A) - 7(D) represents only one way of implementing the present invention and modifications to the disclosed procedure will be readily apparent to those skilled in the art. The additional processing methods necessary for implementing the second embodiment have been described above with reference to the public key processing steps.

The processing sequence control begins at step 2 by fetching a new cell (N). In other words, a new cell is pre-fetched and stored in a buffer and placed in the "pipeline." Once the previous cell (N-1) has completed processing at step 4, the new cell (N) is loaded into the 3DES crypto unit and the MD5/SHA1

authentication unit at step 6. If there are more than two outstanding PCI writes pending, the processing waits until only two or less PCI writes are left to perform (step 8). This ensures that the crypto engines do not outpace the ability of the PCI bus and system to handle the output. Depending on the system, the number of PCI writes that are pending can be adjusted to suit the performance issues of a particular system, interface and bus design.

The crypto processing and authentication processing are then performed in parallel at steps 10 - 16. First, the crypto processing is started for the current cell (N), at step 10, and then a PCI write is queued up at step 12 for the previous cell (N-1) that has just completed processing. Meanwhile, authentication processing is delayed if two PCI writes are pending (step 14). Then the authentication processing for the current cell (N) is started at step 16. If the authentication for the packet is now completed with the processing of the last cell (step 18), the outer HMAC state is written and the outer authentication processing started (step 20). As is known in the art, Hashed Message Authentication Codes (HMAC) use secret keys to prevent someone from changing the data and signing the packet. Since the authentication algorithms are known publicly, the HMAC codes are used to provide greater packet security.

If the authentication output is ready (step 22), a new PCI write is queued up at step 24. If, however, the current cell is the first cell into the authentication unit (step 26), an inner HMAC state is written and the inner authentication is started (step 28). If the pre-fetch of the next cell has started, then the inner HMAC state is written and the inner authentication started (step 32), otherwise processing jumps to "D" on Figure 7(C). Following the inner HMAC write, process control returns to "C" on Figure 7(A), beginning with step 4.

At step 34, a determination is made whether the next cell is part of the same packet as the current cell. If it is, the next cell (N+1) is pre-fetched (step 36), and once the pre-fetch has completed (step 38), the processing returns to "C" on Figure 7(A). If however the next cell is not part of the current packet (i.e. the current packet has completed processing), a determination is made at step 40 whether the packets are part of the same Master Command Record (MCR). As discussed previously, the system may place multiple packets into a single MCR in order to reduce the system overhead, by enabling multiple packets to be processed via a single PCI write. If the packets are from the same MCR, then the context is fetched for the next packet (step 44). If the packets are from different MCRs, however, the crypto and authentication blocks are first drained, the outputs are written, and the MCR status flags are updated (step 42), before the context for the next packet is obtained.

The first cell of the new packet is pre-fetched at step 46. Once the crypto and authentication processing are finished for the last cell of the current packet (step 48), the crypto and authentication processing modes are changed (step 50), as dictated by the new context. The initial states are loaded (step 52), and the previous packet's output descriptors are then saved (step 54). Processing then returns to "A" on Figure 7(A), and the entire procedure continues until there are no more packets to process.

As described, the processing sequence control is highly pipelined, with overlapping execution of various functions. The fact that the packets are split into fixed-sized cells allows for very efficient control sequence processing. Thus, the present invention provides greater throughput than prior art designs. As shown in Figure 8, for example, the first embodiment of the present invention described

above (uBSec ™ 5501 @ 60 MHz), has much greater throughput than a comparable prior art design (Hi/fn ™ 7751 @ 66 MHz). Not only is the present invention faster, it is able to obtain the performance increase without requiring any additional attached memory, as required by the Hi/fn ™ chip.

Those skilled in the art will appreciate that various adaptations and modifications of the just-described preferred embodiments can be configured without departing from the scope and spirit of the invention. For example, other crypto engines may be used, different system interface configurations may be used, or modifications may be made to the cell processing procedure. Therefore, it is to be understood that, within the scope of the appended claims, the invention may be practiced other than as specifically described herein.

CLAIMS

What is claimed is:

1. A method for accelerating cryptography processing of data packets, the method comprising:

splitting an incoming packet into at least one fixed-sized cell;

processing the at least one fixed-sized cell; and

recombining the at least one fixed-sized cell associated with the incoming packet into a processed data packet.

2. The method of Claim 1, wherein if the incoming packet is larger than a single fixed-sized cell, the incoming packet is split into a plurality of fixed-sized cells, whereas if the incoming packet is smaller than a single fixed-sized cell, the packet is converted into a single fixed-sized cell.

3. The method of Claim 2, further comprising reading an incoming packet from a system memory before splitting the packet into fixed-sized cells.

4. The method of Claim 3, further comprising writing the processed data packet out to the system memory.

5. The method of Claim 4, further comprising storing the fixed-sized cells in a buffer.

6. The method of Claim 5, further comprising pre-fetching context information associated with the incoming packet and storing the context information in an on-chip context buffer.

7. The method of Claim 6, wherein processing comprises performing at least one cryptographic operation on the fixed-sized cells.

8. The method of Claim 7, wherein the processing comprises performing 3DES-CBC encryption/decryption and MD5/SHA1 authentication/digital signature processing on the fixed-sized cells.

9. The method of Claim 7, wherein the processing comprises Diffie-Hellman/RSA/DSA public key processing.

10. The method of Claim 8, wherein for in-bound packets, the cells are first authenticated and then decrypted in parallel fashion and for out-bound packets, the cells are first encrypted then authenticated, in pipelined fashion.

11. The method of Claim 10,wherein the cryptographic processing operations are performed in parallel.

12. A method for accelerating IPSec cryptography processing of IP packets, the method comprising:

splitting an incoming IP packet into a plurality of fixed-sized cells, wherein if the incoming IP packet is smaller than a predetermined fixed size, the IP packet is converted into a single fixed-sized cell;

processing the fixed-sized cells with a 3DES-CBC encryption/decryption unit and an MD5/SHA1 authentication/digital signature unit; and

recombining the fixed-sized cells into a processed IP packet.

13. The method of Claim 12, further comprising reading an incoming IP packet from a system memory before splitting the packet into fixed-sized cells.

14. The method of Claim 12, further comprising writing the processed IP packet out to the system memory.

15. The method of Claim 14, further comprising storing the fixed-sized cells in an on-chip buffer.

16. The method of Claim 15, further comprising pre-fetching context information associated with the incoming packet and storing the context information in a context buffer.

17. The method of Claim 16, wherein for in-bound packets, the cells are first authenticated and then decrypted in parallel fashion and for out-bound packets, the cells are first encrypted then authenticated, in pipelined fashion.

18. The method of Claim 17,wherein the cryptographic processing operations are performed in parallel.

19. The method of Claim 18, wherein multiple packets are combined into a single record and are sent for processing by a system controller with a single bus write command.

20. A cryptography acceleration chip comprising:

a packet splitting unit, in which incoming packets are split into fixed-sized cells;

at least one cryptography processing block connected to receive the fixed-sized cells from the packet splitting unit; and

a control processor that sequences the processing of the fixed-sized cells through the at least one cryptography processing block without requiring any attached local memory.

21. The cryptography acceleration chip of Claim 20, further comprising an input buffer for holding the input packets read from a system memory.

22. The cryptography acceleration chip of Claim 21, further comprising an external bus interface.

23. The cryptography acceleration chip of Claim 22, further comprising an output buffer.

24. The cryptography acceleration chip of Claim 23, further comprising a context pre-fetch buffer and a current context buffer.

25. The cryptography acceleration chip of Claim 24, wherein the packet splitting unit comprises a data align barrel shifter.

26. The cryptography acceleration chip of Claim 25, further comprising a 3DES-CBC encryption/decryption unit and an MD5/SHA1 authentication/digital signature unit.

27. The cryptography acceleration chip of Claim 26, further comprising a Diffie-Hellman/RSA/DSA public key processing unit.

28. The cryptography acceleration chip of Claim 27, further comprising a random number generator.

29. The cryptography acceleration chip of Claim 28, further comprising a register files unit.

30. The cryptography acceleration chip of Claim 26, for in-bound packets, the cells are first authenticated and then decrypted in parallel fashion and for out-bound packets, the cells are first encrypted then authenticated, in pipelined fashion.

31. An IPSec cryptography acceleration chip comprising:

an external system bus interface unit;

a packet splitting unit, in which incoming packets are split into fixed-sized cells;

a 3DES-CBC encryption/decryption unit and an MD5/SHA1 authentication/digital signature unit connected to receive the fixed-sized cells from the packet splitting unit;

a first FIFO input buffer connected to the 3DES-CBC unit;

a second FIFO input buffer connected to the MD5/SHA1 unit;

a first FIFO output buffer connected to the 3DES-CBC unit;

a second FIFO output buffer connected to the MD5/SHA1 unit;

a pre-fetch context buffer;

a current context buffer; and

a control processor that sequences the processing of the fixed-sized cells through the 3DES-CBC encryption/decryption unit and the MD5/SHA1 authentication/digital signature unit.

32. The IPSec cryptography acceleration chip of Claim 31, further comprising:

a DH/RSA/DSA public key processing unit;

a random number generator; and

a register files unit.

33. A network communication device comprising:

a central processing unit;

a system memory;

a network interface unit;

a cryptography acceleration chip comprising:

a packet splitting unit, in which incoming packets are split into fixed-sized cells;

at least one cryptography processing block connected to receive the fixed-sized cells from the packet splitting unit; and

a control processor that sequences the processing of the fixed-sized cells through the at least one cryptography processing block without requiring any attached local memory; and

an internal bus that connects the central processing unit, the system memory, the network interface unit, and the cryptography acceleration chip.

34. A method for sequencing fixed-sized cells in a cryptography acceleration chip, wherein incoming data packets are split into fixed-sized cells, the method comprising:

pre-fetching a next cell for processing;

waiting until a previous cell has finished processing;

loading the next cell into a cryptography processing unit;

waiting until less than a predetermined number of system bus writes are pending; and

starting the cryptography processing on a current cell and queuing up a write for the previous cell.

35. The method of Claim 34, wherein the cryptography processing comprises performing both encryption/decryption and authentication in parallel.

36. The method of Claim 35, further comprising writing an outer HMAC code, if the current cell is the last cell to be authenticated in a current packet.

37. The method of Claim 36, further comprising writing an inner HMAC code, if the current cell is the first cell in a new packet.

38. The method of Claim 37, further comprising pre-fetching a next cell, if the next cell is from a same packet as a current cell, otherwise, determining if a new packet is part of a same Master Command Record (MCR) as the current packet.

39. The method of Claim 38, further comprising pre-fetching a new context and a new cell, if the new packet is part of the same MCR as the current packet, otherwise, draining the cryptography processing blocks, writing an output, updating status MCR status information, and then pre-fetching a new context and a new cell.

40. The method of Claim 39, wherein the encryption/decryption is performed by a 3DES-CBC unit and the authentication is performed by a MD5/SHA1 unit in order to implement IPSec processing.

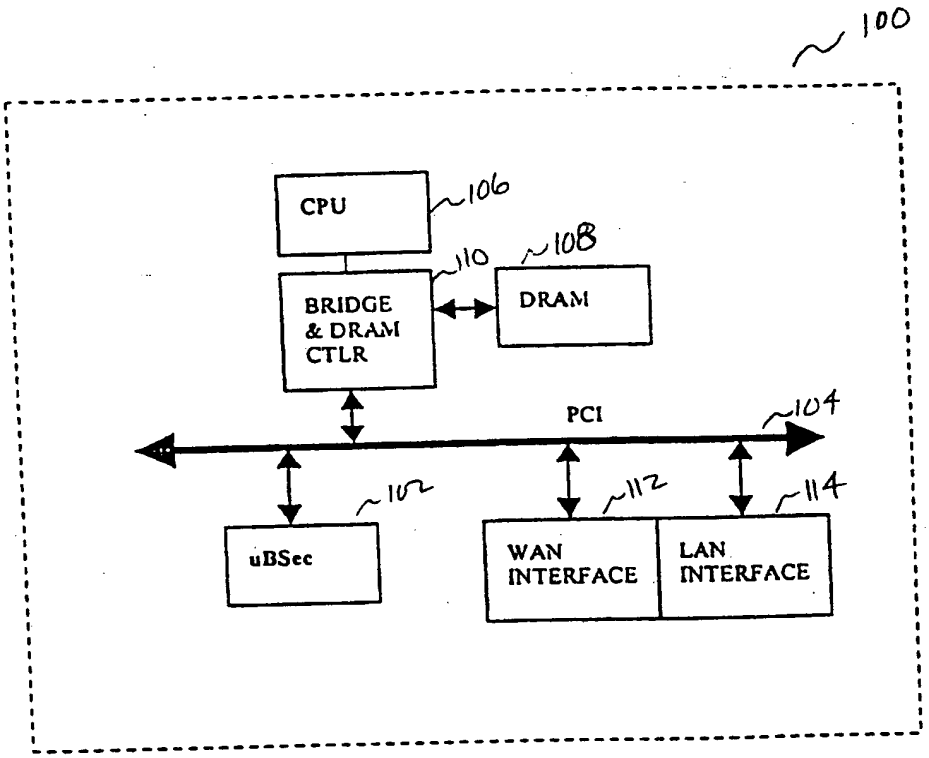


Figure 1

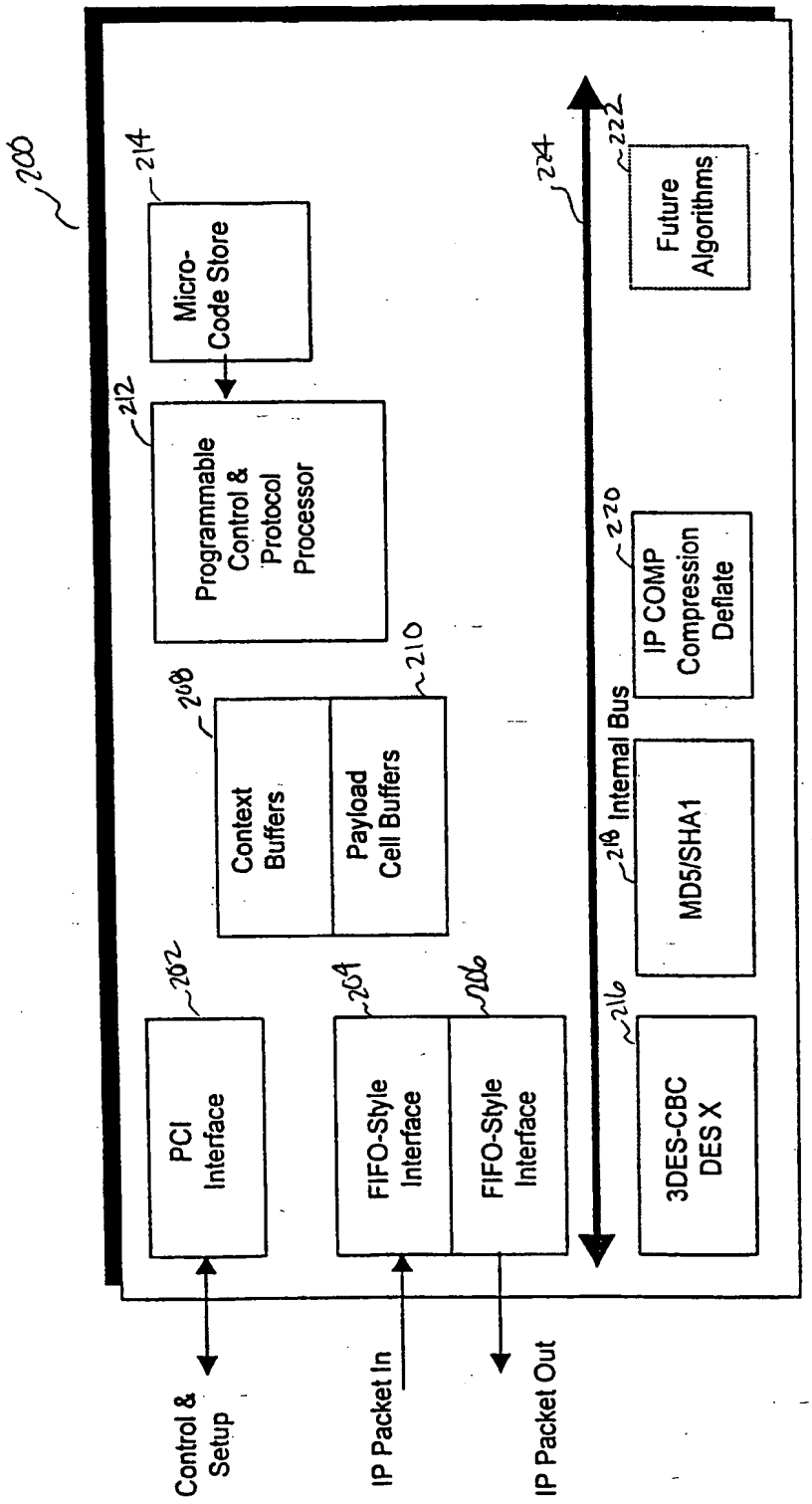


Figure 2

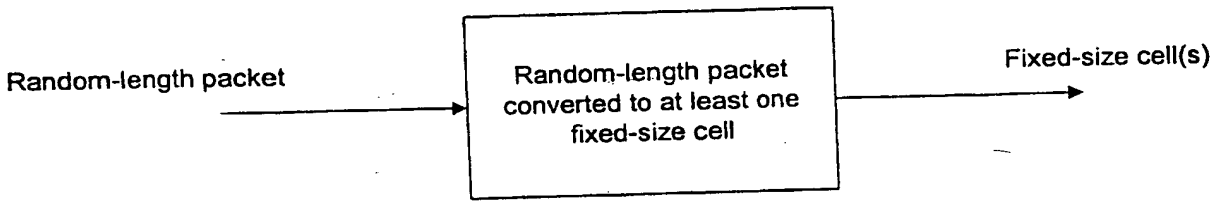

Random-length packet
Random-length packet converted to at least one fixed-size cell
Fixed-size cell(s)

Figure 3

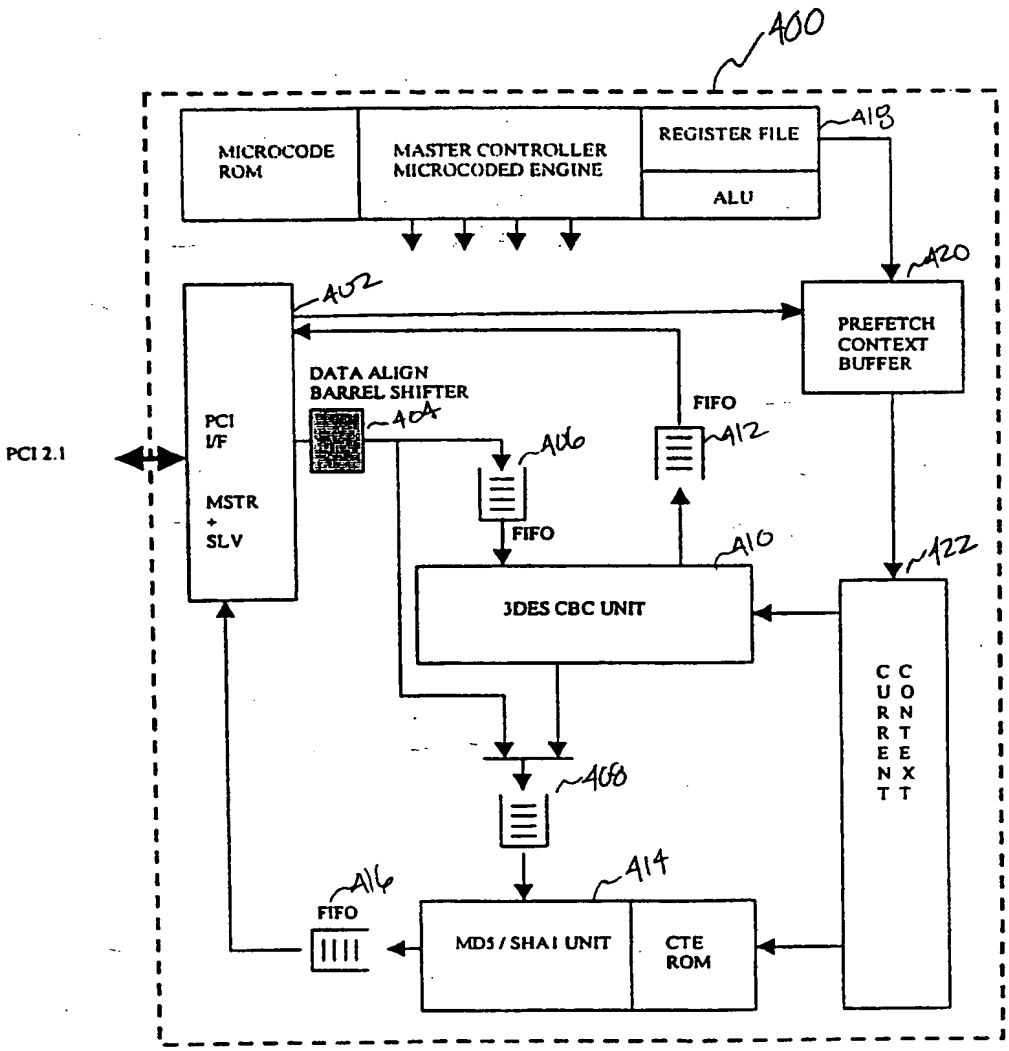


Figure 4

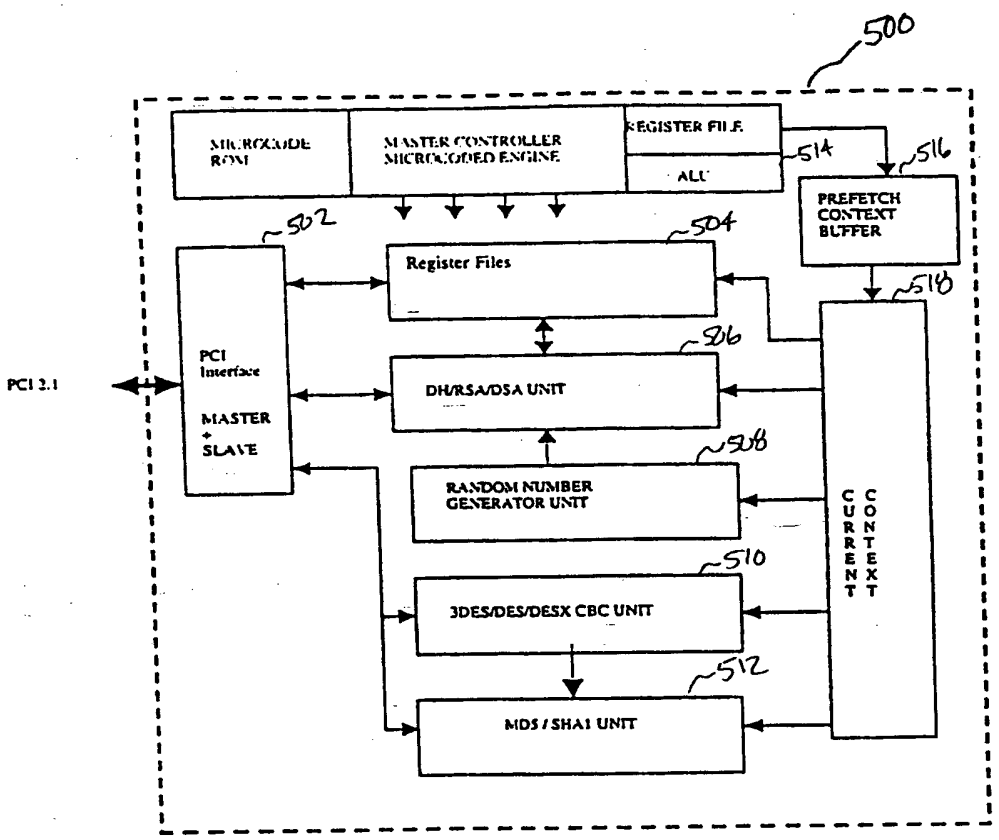

500
MICROCODE ROM
MASTER CONTROLLER MICROCODED ENGINE
REGISTER FILE
ALU
514
516
PREFETCH CONTEXT BUFFER
502
504
Register Files
518
PCI 2.1
PCI Interface
MASTER + SLAVE
506
DH/RSA/DSA UNIT
508
RANDOM NUMBER GENERATOR UNIT
CURRENT CONTEXT
510
3DES/DES/DESX CBC UNIT
512
MD5 / SHA1 UNIT

Figure 5

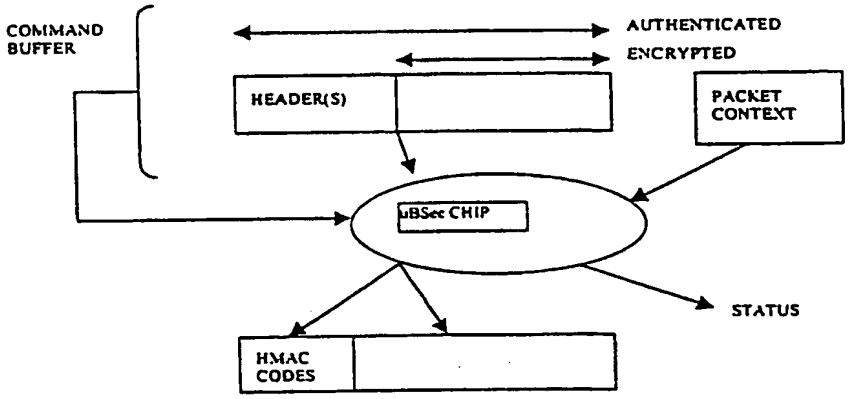

COMMAND BUFFER
AUTHENTICATED
ENCRYPTED
HEADER(S)
PACKET CONTEXT
uBSec CHIP
STATUS
HMAC CODES

Figure 6

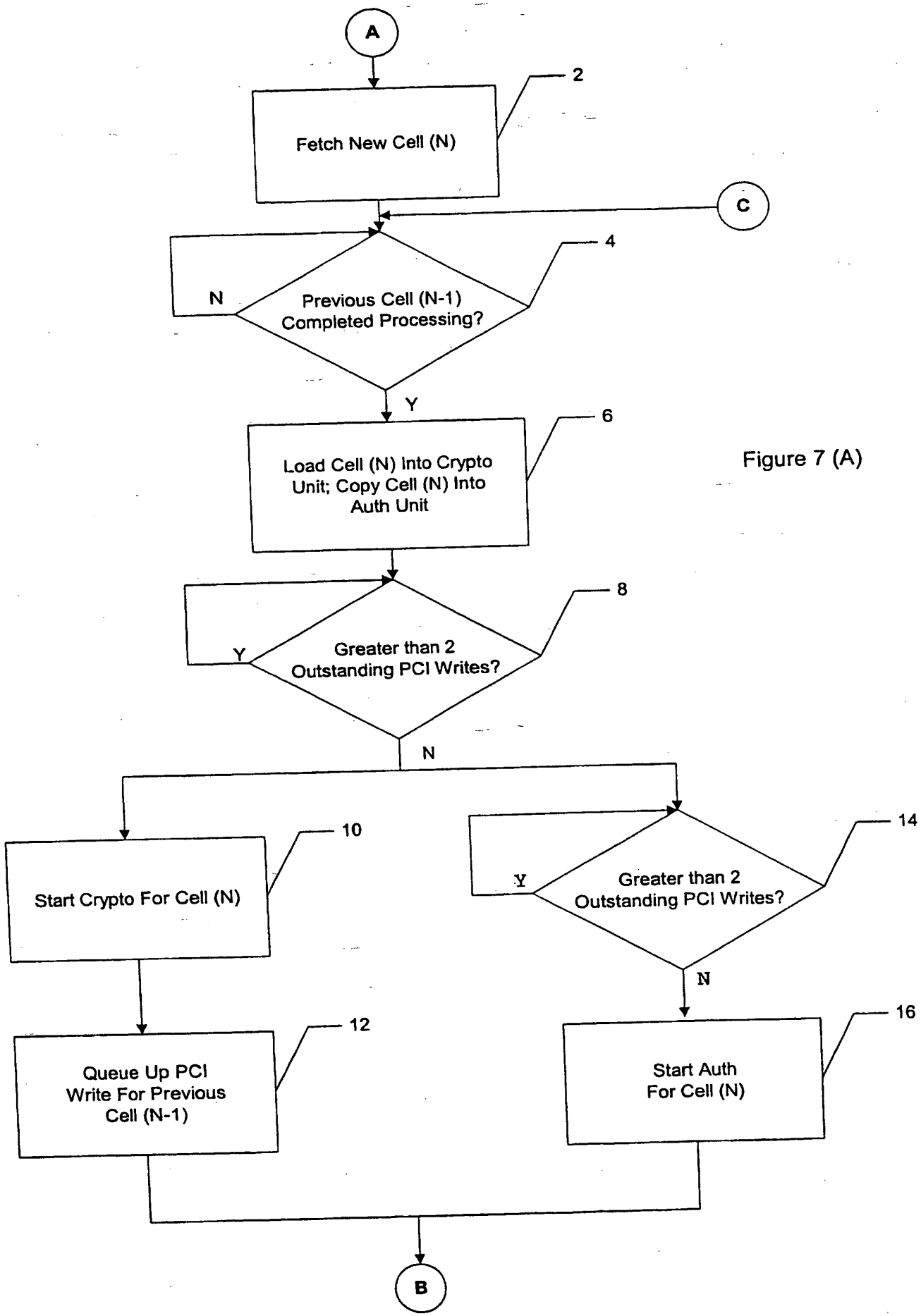

A
Fetch New Cell (N)
2
C
Previous Cell (N-1) Completed Processing?
4
N
Y
Load Cell (N) Into Crypto Unit; Copy Cell (N) Into Auth Unit
6
Greater than 2 Outstanding PCI Writes?
8
Y
N
Start Crypto For Cell (N)
10
Queue Up PCI Write For Previous Cell (N-1)
12
Greater than 2 Outstanding PCI Writes?
14
Y
N
Start Auth For Cell (N)
16
B

Figure 7 (A)

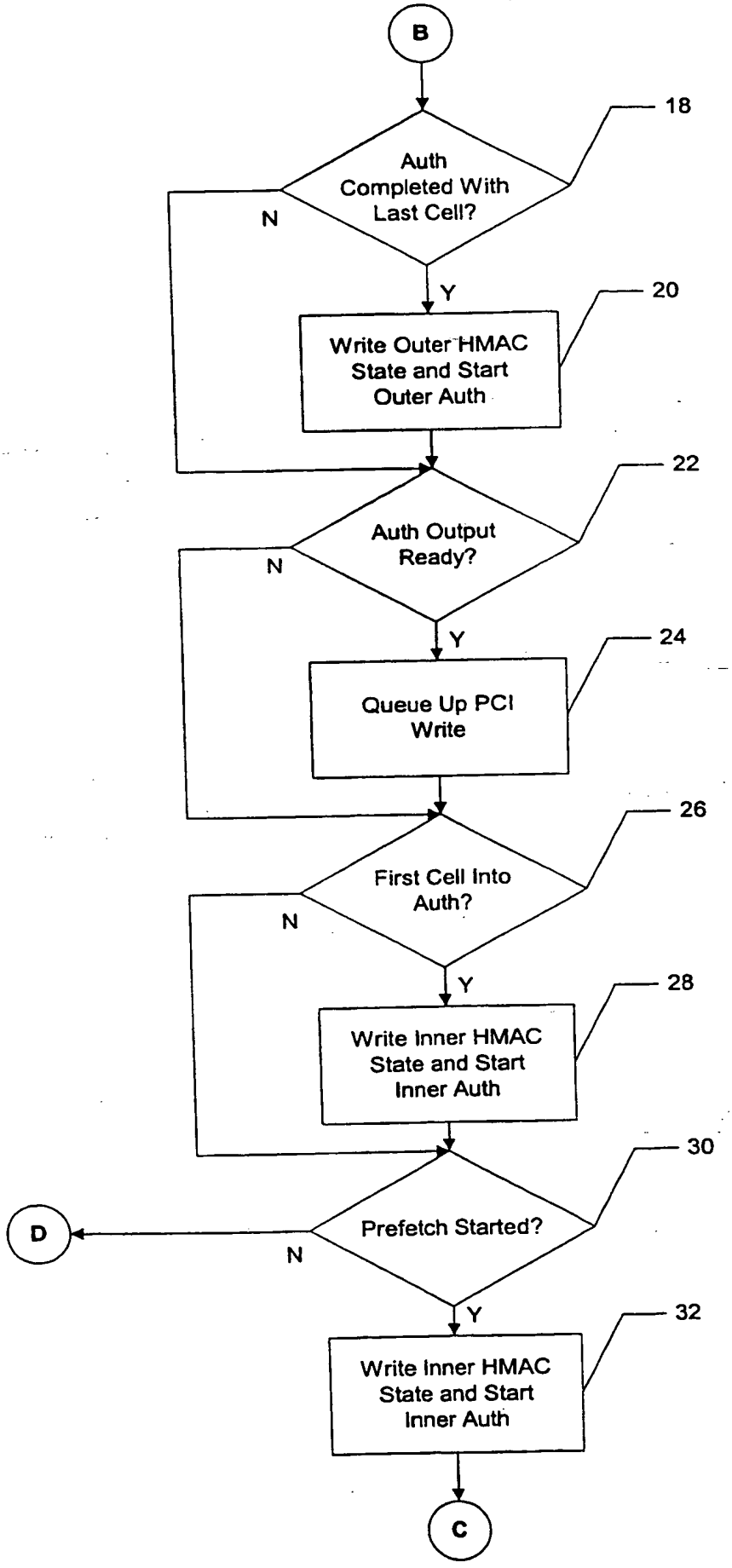


Figure 7 (B)

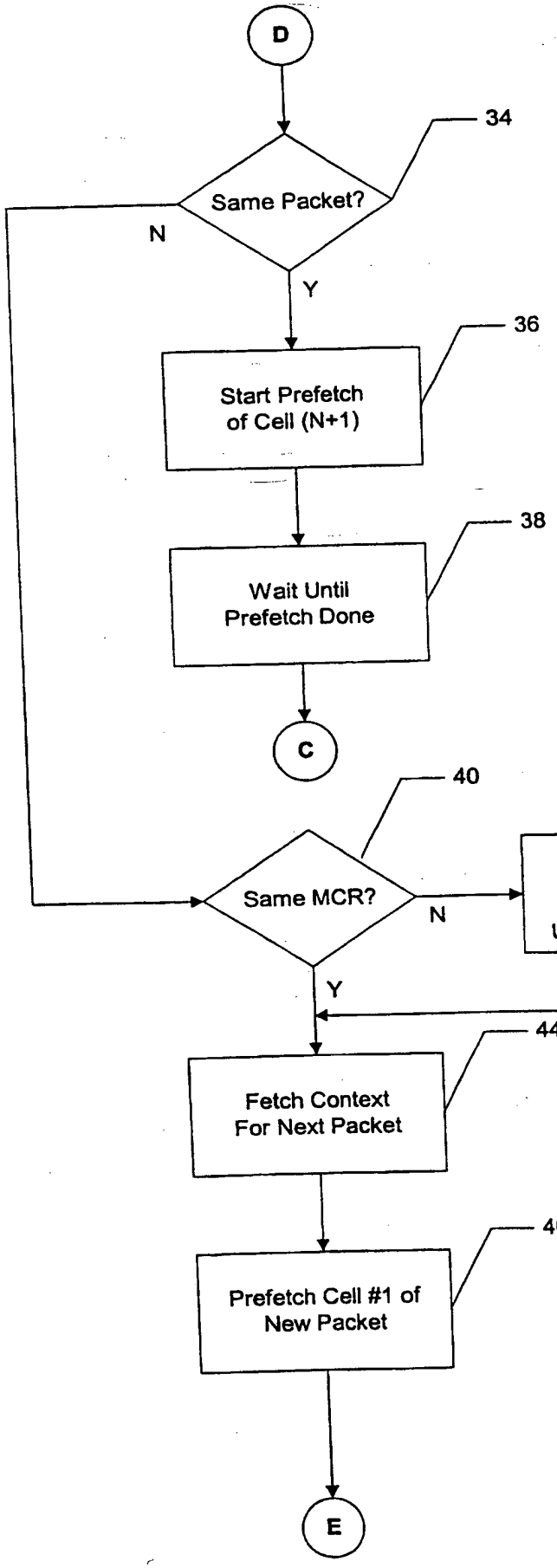

D
34
Same Packet?
N
Y
36
Start Prefetch of Cell (N+1)
38
Wait Until Prefetch Done
C
40
Same MCR?
N
42
Drain Crypto
Drain Auth
Write Outputs
Update MCR Flags
Y
44
Fetch Context For Next Packet
46
Prefetch Cell #1 of New Packet
E

Figure 7 (C)

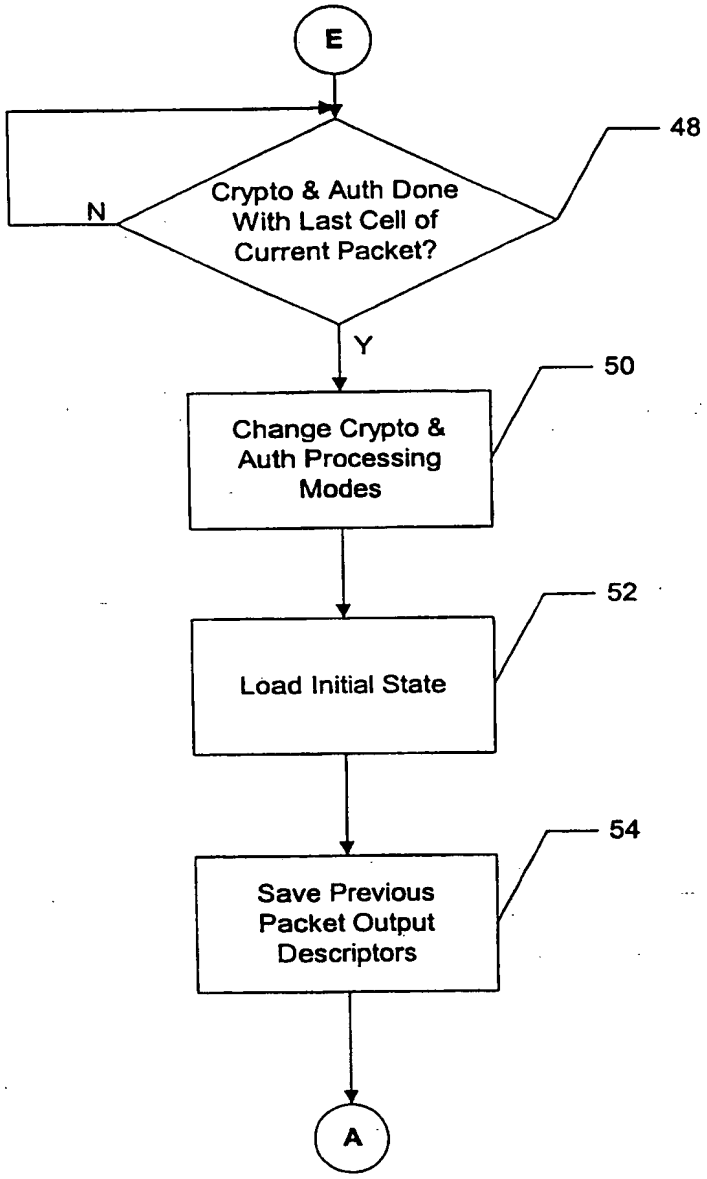


Figure 7 (D)

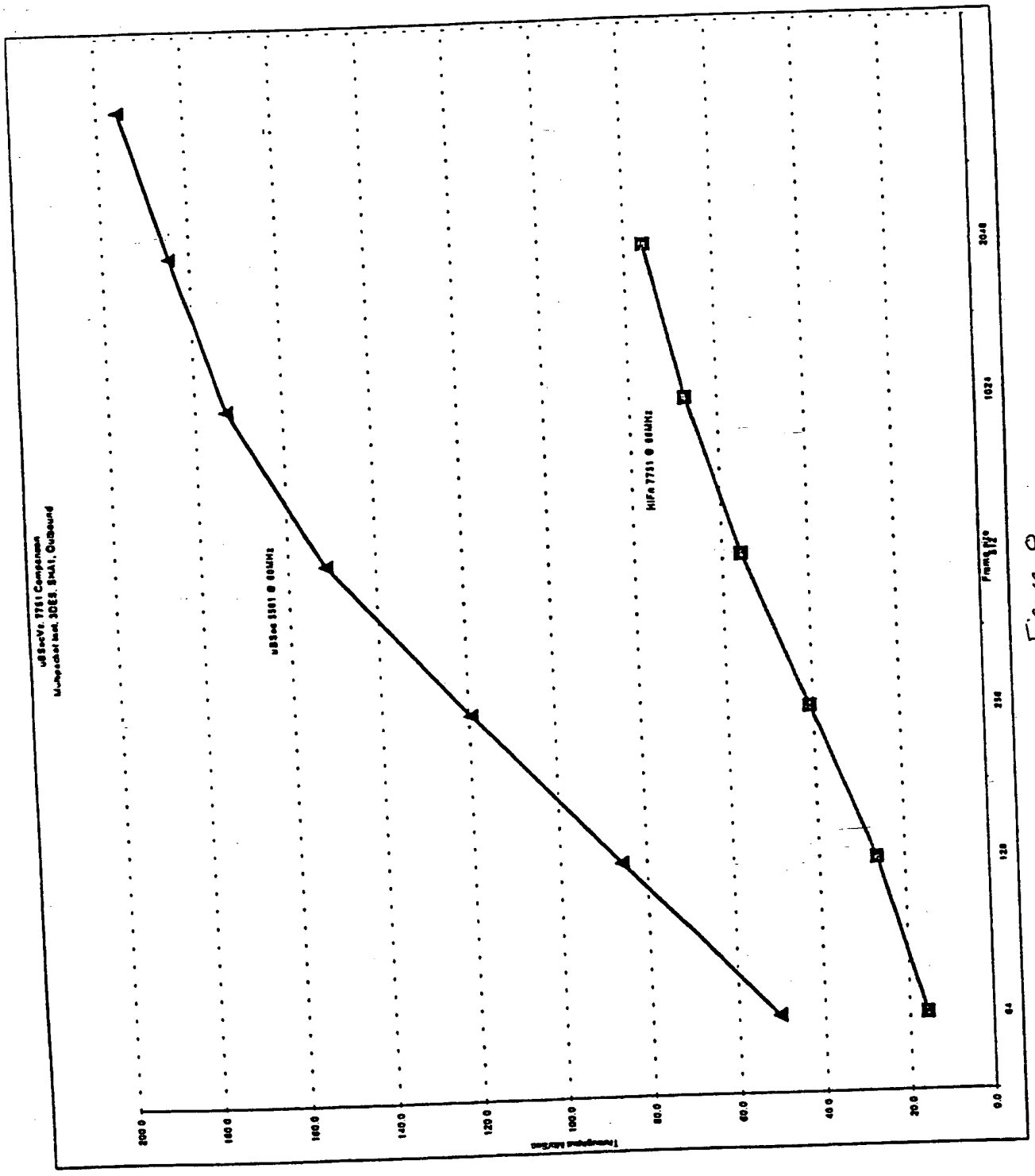

HiFn 7751 @ 66MHz
Throughput kb/sec
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
2048
1024
256
128
64

Figure 8

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 January 2001 (18.01.2001)

PCT

(10) International Publication Number
WO 01/05089 A3

(51) International Patent Classification[7]: H04L 29/06

(21) International Application Number: PCT/US00/18545

(22) International Filing Date: 7 July 2000 (07.07.2000)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/142,870	8 July 1999 (08.07.1999)	US
60/159,012	12 October 1999 (12.10.1999)	US
09/510,486	23 February 2000 (23.02.2000)	US

(71) Applicant *(for all designated States except US)*: **BROADCOM CORPORATION** [US/US]; 16215 Alton Parkway, Irvine, CA 92618 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **KRISHNA, Suresh** [US/US]; 695 S. Knickerbocker Drive #6, Sunnyvale, CA 94087 (US). **OWEN, Christopher** [US/US]; 708 Blossom Hill Road #198, Los Gatos, CA 95032 (US).

(74) Agent: **AUSTIN, James, E.**; Beyer Weaver & Thomas, LLP, P.O. Box 130, Mountain View, CA 94042-0130 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

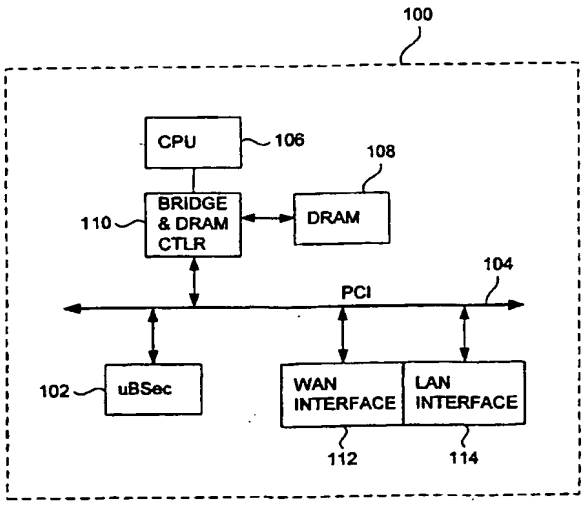
(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*

(88) Date of publication of the international search report:
27 September 2001

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SECURITY CHIP ARCHITECTURE AND IMPLEMENTATIONS FOR CRYPTOGRAPHY ACCELERATION



(57) Abstract: An architecture and a method for a cryptography acceleration is disclosed that allows significant performance improvements without the use of external memory. Specifically, the chip architecture enables "cell-based" processing of random-length IP packets. The IP packets, which may be of variable and unknown size, are split into fixed-size "cells". The fixed-sized cells are then processed and reassembled into packets. The cell-based packet processing architecture of the present invention allows the implementation of a processing pipeline that has known processing throughput and timing characteristics, thus making it possible to fetch and process the cells in a predictable time frame. The architecture is scalable and is also independent of the type of cryptography performed. The cells may be fetched ahead of time (pre-fetched) and the pipeline may be staged in such a manner that attached (local) memory is not required to store packet data or control parameters.

WO 01/05089 A3

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 00/18545

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)
EPO-Internal, IBM-TDB, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	S. KENT, R. ATKINSON: "RFC 2406 - IP Encapsulating Security Payload (ESP) " IETF REQUEST FOR COMMENTS, 'Online! November 1998 (1998-11), XP002163400 Retrieved from the Internet: <URL:http://www.faqs.org/rfcs/frc2406.html > 'retrieved on 2001-03-20!	1,2,12, 20
Y	page 4-5, sec 2.4 Padding (for Encryption)	13-19
A	page 7, sec 3.2.2 Authentication Algorithms	31,33,34
	--- -/--	

[X] Further documents are listed in the continuation of box C.

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
21 March 2001	04/04/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31 651 epo nl, Fax: (+31-70) 340-3016	Bertolissi, E

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int ional Application No
PCT/US 00/18545

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
X	SHOLANDER P ET AL: "THE EFFECT OF ALGORITHM-AGILE ENCRYPTION ON ATM QUALITY OF SERVICE" GLOBAL TELECOMMUNICATIONS CONFERENCE (GLOBECOM),US,NEW YORK, IEEE, 3 November 1997 (1997-11-03), pages 470-474, XP000737578 ISBN: 0-7803-4199-6	1-4
A	figure 1 page 473, col 1, lines 6-10	12,20, 31,33,34

X	PIERSON L G ET AL: "CONTEXT-AGILE ENCRYPTION FOR HIGH SPEED COMMUNICATION NETWORKS" COMPUTER COMMUNICATIONS REVIEW,US,ASSOCIATION FOR COMPUTING MACHINERY. NEW YORK, vol. 29, no. 1, January 1999 (1999-01), pages 35-49, XP000823872 ISSN: 0146-4833	1-11, 20-27,30
Y	page 37, par 2 page 38, lines 4-9 page 40, par 3 figure 2 pag 43, sec 5.1.2.3 ATM Cell Format pag 44, par 4 pag 47, par 2	13-19, 28,29, 31-40

X	US 5 796 836 A (MARKHAM THOMAS R) 18 August 1998 (1998-08-18) column 12, line 2 - line 16 column 14, line 32 -column 38 figures 8A,,12A	1-11, 20-27,30

Y	ANALOG DEVICES: "Analog Devices and IRE announce first DSP-based internet security system-on-a-chip (ADSP2131)" ANALOG DEVICES PRESS RELEASE, 'Online! 19 January 1999 (1999-01-19), XP002163285 Nordwood, Ma whole document	28,29, 31,32

A	ANALOG DEVICES: "ADSP2141 SafeNetDPS USER'SMANUAL, revision 6" ANALOG DEVICES TECHNICAL SPECIFICATIONS, March 2000 (2000-03), XP002163401 figure 1; pages 1-4, page 34, figure 10	2,29,31

	-/--	

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Int. ional Application No
PCT/US 00/18545

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication,where appropriate, of the relevant passages	Relevant to claim No.
Y	3COM: "3Com launces new Era of Network Connectivity" 3COM PRESS RELEASE, 'Online! 14 June 1999 (1999-06-14), XP002163286 Santa Clara, CA Retrieved from the Internet: <URL:http://www.3com.com/news/releases/pr99/jun1499a.html> 'retrieved on 2001-03-20! whole document ---	33
Y	C. MADSON, R. GLENN: "RFC 2403 - The use of HMAC-MD5-96 within ESP and AH" IETF REQUEST FOR COMMENTS, 'Online! November 1998 (1998-11), XP002163402 Retrieved from the Internet: <URL:http://www.faqs.org/rfcs/frc2403.html> 'retrieved on 2001-03-20! page 1, 1. Introduction -----	34-40

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/18545

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5796836 A	18-08-1998	AU 5542896 A	07-11-1996
		EP 0821853 A	04-02-1998
		JP 10508450 T	18-08-1998
		WO 9633564 A	24-10-1996

Form PCT/ISA/210 (patent family annex) (July 1992)